

Rancang Bangun Sistem Manajemen Palang Parkir AI-IoT Otomatis Menggunakan YOLO & OCR

Maria Angelica Vinesytha Chandrawan¹, Edgar Tanamal², Antonius Indra Dharma Prasetya³, Kasmir Syariati⁴, Citra Suardi^{5*}

^{1,2,3,4,5} PSDKU Informatika, Universitas Ciputra, Surabaya, East Java, Indonesia

e-mail: ¹mangelica01@student.ciputra.ac.id , ²etanamal@student.ciputra.ac.id ,

³aindradharna@student.ciputra.ac.id , ⁴kasmir.syariati@ciputra.ac.id,

⁵Citra.suardi@ciputra.ac.id

* Corresponding author

Abstrak

Manusia dan peradabannya tidak terlepas dari perkembangan teknologi. *Internet of Things* (IoT) menjadi salah satu jenis teknologi berkembang yang dapat membantu dan memenuhi kebutuhan sehari - hari. IoT tersebut dapat digunakan dalam berbagai bidang. Salah satunya dalam bidang transportasi yaitu sistem pelayanan parkir. Permasalahan yang sering dijumpai ketika berkunjung ke suatu tempat khusus seperti mall, apartemen, atau kantor - kantor pada hari libur atau hari raya yaitu pengendara merasa sulit mencari tempat parkir. Berdasarkan uraian tersebut pada penelitian ini kami membuat rancang bangun sistem manajemen palang berbasis AI-IoT yang otomatis dengan menggunakan YOLO dan OCR. Dari penelitian ini kami memperoleh hasil yang memuaskan terhadap penggunaan YOLOv4 dan OCR untuk mendeteksi plat kendaraan dengan tingkat akurasi yang tinggi yaitu 98,75%.

Kata kunci: *Internet of Things, Artificial Intelligence, YOLO, OCR, Parking*

Abstract

Humans and civilization cannot be separated from technological developments. The Internet of Things (IoT) is one type of developing technology that can help and meet everyday needs. The IoT can be used in various fields. One of them in the field of transportation is the parking service system. The problem that is often encountered when visiting a special place such as a mall, apartment, or offices on holidays or holidays is that drivers find it difficult to find a parking space. Based on this description, in this study we designed an automated AI-IoT-based crossbar management system using YOLO and OCR. From this study we obtained satisfactory results regarding the use of YOLOv4 and OCR to detect vehicle plates with a high accuracy rate of 98.75%.

Keywords: *Internet of Things, Artificial Intelligence, YOLO, OCR, Parking*

1. Introduction

Perkembangan teknologi informasi belakangan ini telah menyebar luas sampai ke berbagai sektor seperti kesehatan, transportasi, pendidikan, bahkan di dalam kehidupan sehari – hari tak jarang lagi kita jumpai teknologi informasi yang semakin canggih dimana hal tersebut dapat mempermudah hidup kita. Selain itu, teknologi informasi mampu melakukan pengolahan dan pemrosesan data dengan kapasitas yang banyak. Salah satu teknologi yang berkembang untuk menyelesaikan permasalahan kompleks dalam kehidupan masyarakat adalah *Artificial Intelligence of Things* (AIoT).

AIoT merupakan gabungan dari *Artificial Intelligence* (AI) dan *Internet of Things* (IoT) untuk mencapai operasi IoT yang lebih efisien, meningkatkan interaksi manusia - mesin, dan meningkatkan manajemen data dan analitik [1][2][3]. AIoT memiliki tujuan untuk memperluas manfaat konektivitas internet yang tersambung secara terus menerus, yang memungkinkan kita untuk menghubungkan mesin, peralatan, dan benda fisik lainnya. Menurut laporan Ketua Umum Asosiasi IoT Indonesia (ASIOTI), Teguh Prasetya pada 2025 pasar IoT di Indonesia diprediksi mampu bisa mencapai 40 miliar dolar AS atau Rp 572,7 triliun, dengan 678 juta perangkat IoT yang sudah terhubung [5]. AIoT membantu meningkatkan penyelesaian berbagai masalah seperti pengiriman produk yang efisien dan peningkatan kualitas produk [6].

Permasalahan yang sering dijumpai ketika berkunjung di suatu tempat khususnya seperti, mall, apartemen, atau kantor yaitu permasalahan fasilitas umum (tempat parkir). Para pengendara merasa kesulitan dalam mencari tempat parkir terkhusus pada hari liburan, yang mana tempat parkir menumpuk, tidak tertata rapi karena tidak dilengkapi akses informasi slot tempat parkir. Kesulitan dalam mencari tempat parkir mengakibatkan terjadinya beberapa masalah, mulai dari kemacetan, pengendara merasa tidak nyaman, bahkan meningkatnya polusi udara. Permasalahan kemacetan terjadi dikarenakan pengendara merasa sulit untuk mendapatkan parkir sehingga mengakibatkan antrian dalam mencari tempat parkir. Pengendara juga merasa tidak nyaman atau risih karena pengendara merasa membuang waktu yang cukup banyak hanya untuk mencari tempat parkir. Selain itu, jika pengendara sulit untuk mendapatkan tempat parkir maka pengendara harus berputar beberapa kali hingga mendapatkan tempat parkir, di mana akibatnya pasti banyak membuang bahan bakar sehingga dapat meningkatkan polusi udara.

Pengembangan teknologi akses tempat parkir merupakan kebutuhan pada sistem parkir secara otomatis. Akses sistem informasi yang otomatis dapat menjadi media prasarana yang mempermudah pengelolaan parkir dengan rapi. Adapun teknologi yang digunakan adalah sistem otomatis akses informasi mulai dari lokasi (mall dan lantai) hingga jumlah keseluruhan tempat parkir yang kosong maupun terisi. Sehingga pelayanan parkir menjadi sangat informatif dan real time dalam menampilkan slot-slot parkir yang berada di mall dan lantai yang masih tersedia.

Berdasarkan uraian yang telah dijelaskan, solusi yang kami kembangkan yaitu "Sistem Kendali Tempat Parkir Berbasis *Artificial Intelligence of Things* (AIoT)". Sistem tersebut dapat diakses oleh *user* dengan layanan internet melalui *smartphone* android dengan melalui perantara *firebase* untuk mempermudah pengendara mengetahui akses informasi lokasi yang akan dijadikan tempat parkir. Selain itu, pada *project* ini juga menggunakan YOLO sebagai pendeteksi objek dalam hal ini plat kendaraan dan teknologi OCR untuk melakukan pencatatan plat mobil sehingga palang parkir yang menggunakan motor servo akan otomatis terbuka saat web kamera membaca plat mobil yang berada di depan palang dan otomatis kembali tertutup saat sensor mendeteksi bahwa kendaraan sudah melewati palang parkir.

2. Research Method

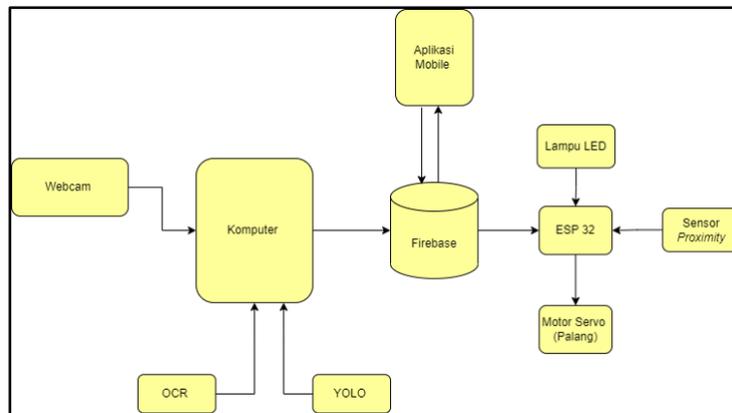
Perancangan sistem yang dilakukan adalah *prototype* pencatatan jumlah kendaraan yang masuk ke area parkir dan palang pintu parkir yang terbuka secara otomatis. Pada sistem ini menggunakan web kamera dan mikrokontroler. Pembahasan rancangan sistem dimulai dari pembahasan diagram blok sistem, perancangan perangkat keras, dan perancangan perangkat lunak.

2.1. Diagram Blok Sistem

Diagram blok sistem pencatatan jumlah kendaraan yang masuk ke area parkir dan palang pintu parkir yang terbuka secara otomatis ditunjukkan pada Gambar 1. Diagram blok terdiri atas webcam, sensor *proximity*, motor servo, mikrokontroler, dan power supply.

Web kamera berfungsi untuk membaca plat nomor yang ada pada kendaraan. Sensor Proximity digunakan sebagai pembaca pergerakan kendaraan melewati sensor. Data hasil pembacaan sensor dikirim ke mikrokontroler untuk diproses agar palang pintu tertutup. Modul mikrokontroler yang digunakan adalah ESP32. Bagian output sistem terdiri dari motor servo yang berfungsi sebagai motor penggerak palang pintu. Bagian utama pada sistem ini adalah web kamera yang berfungsi untuk membaca plat nomor kendaraan yang masuk ke dalam area parkir.

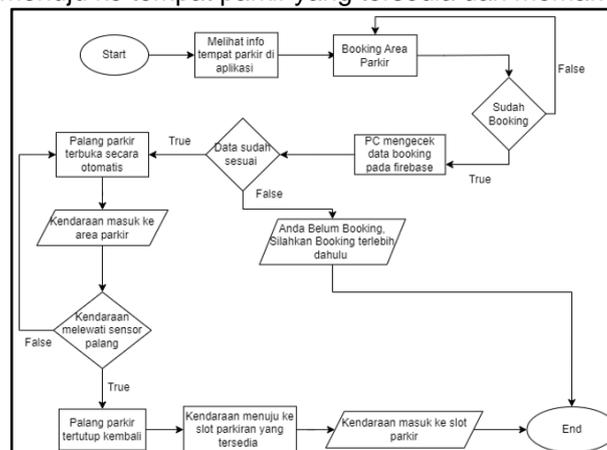
Sistem juga dilengkapi dengan aplikasi Python yang berfungsi mengubah gambar menjadi teks, menyimpan teks yang menggunakan aplikasi *firebase* untuk menyimpan data plat nomor yang masuk ke dalam area parkir, dan mencocokkan plat kendaraan pada pintu masuk.



Gambar 1. Diagram Blok Sistem.

2.2. Prinsip Kerja Sistem

Sistem mulai bekerja saat pengendara melihat informasi area parkir yang terdapat pada aplikasi android yang dibuat. Setelah pengendara melihat informasi dan memilih area parkir yang ia tuju, pengendara mem-*booking area* parkir yang masih tersedia. Kemudian, sistem akan mengecek apakah data yang di-*input* oleh pengendara sudah sesuai dengan data yang dibaca oleh webcam. Jika data yang dicek sudah sesuai, maka palang parkir akan terbuka secara otomatis, sedangkan jika data tidak sesuai, maka sistem membaca bahwa pengendara belum memesan area parkir. Setelah, palang parkir terbuka, kendaraan masuk ke area parkir. Sistem akan menerima data dari sensor yang terdapat di sekitar palang parkir untuk menutup palang parkir secara otomatis. Apabila data yang diterima bahwa kendaraan sudah melewati sensor, palang otomatis akan tertutup, sedangkan jika kendaraan belum melewati sensor, palang parkir akan terbuka terus - menerus. Kendaraan menuju ke tempat parkir yang tersedia dan memarkir kendaraan.



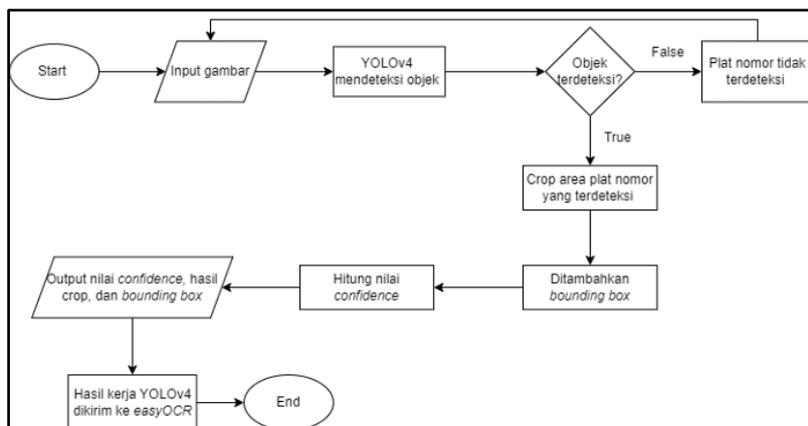
Gambar 2. Flowchart Kerja Sistem Area Parkir.

2.3. Metode

2.3.1. You Only Look Once (YOLO)

YOLO (You Only Look Once) dan CNN (Convolutional Neural Network) adalah dua konsep penting dalam bidang visi komputer dan deteksi objek. CNN adalah sebuah jenis arsitektur jaringan saraf tiruan yang terinspirasi oleh organisasi visual manusia. Arsitektur CNN memiliki lapisan konvolusi pada part *Feature Learning* [4] yang memungkinkannya untuk secara efektif mengenali pola dan fitur dalam data gambar. YOLO memanfaatkan arsitektur CNN untuk mengenali fitur-fitur penting dalam gambar, dan kemudian menggunakan lapisan terakhirnya untuk melakukan prediksi objek dan kotak pembatas.

Dalam penelitian ini, sistem YOLO bekerja pertama kalinya pada saat webcam membaca bahwa terdapat kendaraan. Selanjutnya, kendaraan beserta plat nomornya diterima oleh sistem YOLO sehingga sistem akan mendeteksi objek tersebut. Sistem akan mengecek apakah objek dapat terdeteksi atau tidak. Jika objek dapat terdeteksi maka area plat nomor kendaraan yang terdeteksi akan di crop. Jika objek tidak dapat terdeteksi maka sistem tidak dapat mendeteksi area plat nomor kendaraan. Setelah itu, pada tepi gambar plat nomor kendaraan ditambahkan *bounding box* (garis berwarna merah) serta dihitungnya nilai *confidence* untuk hasil tersebut. Berdasarkan kerja sistem YOLOv4, maka sistem menghasilkan *output* gambar plat nomor kendaraan yang telah di-*crop*, terdapat *bounding box*, dan juga nilai *confidence*. Kemudian, hasil dari sistem YOLOv4 dikirim ke *easyOCR* untuk melanjutkan kerja sistem selanjutnya.

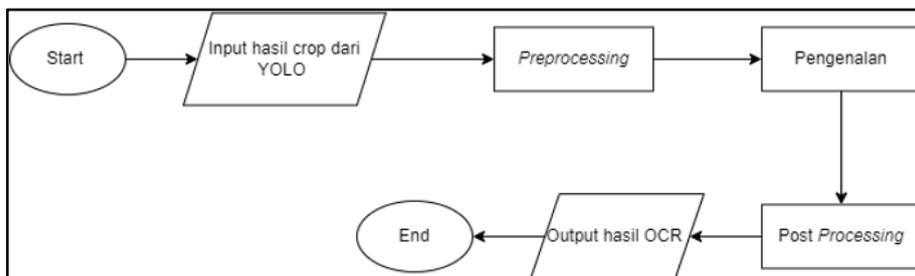


Gambar 3. Flowchart Kerja YOLOv4.

2.3.2. Optical Character Recognition (OCR)

Optical Character Recognition (OCR) adalah teknik yang memungkinkan untuk melakukan pengenalan karakter secara otomatis [7][8]. OCR biasanya terdiri dari pemindaian optik, lokalisasi, segmentasi, preprocessing, ekstraksi fitur, klasifikasi, dan postprocessing [9], pemindaian optik dilakukan dengan memindai gambar. Biasanya OCR melakukan konversi gambar bertingkat menjadi gambar hitam putih. Menurut Stefenon et al. Proses mengubah gambar menjadi hitam putih dapat dicapai dengan *thresholding* atau binarisasi menggunakan metode canggih [10]. Berbagai pendekatan yang dapat digunakan pada OCR, yaitu *matrix matching*, *fuzzy logic*, ekstraksi fitur, analisa struktural, dan jaringan syaraf.

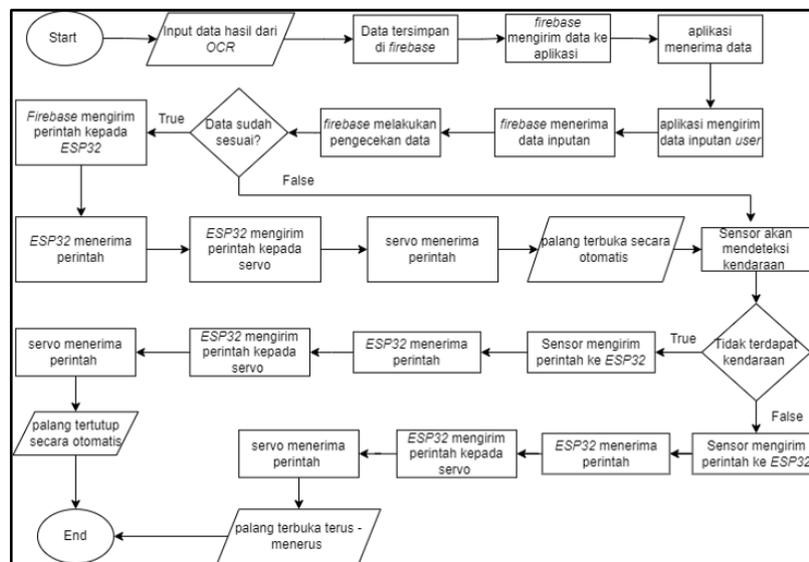
Dalam penelitian ini sistem *easyOCR* akan menerima hasil *crop* yang dikerjakan oleh *openCV* di YOLOv4. Setelah itu, sistem akan melakukan *preprocessing* yang merupakan proses awal dilakukan suatu citra untuk menghilangkan *noise* (bagian – bagian yang tidak diperlukan pada gambar *input* untuk proses selanjutnya). Pada tahap *preprocessing* terdapat beberapa proses yang terjadi yaitu *grayscale* dan *otsu threshold*. *Grayscale* merupakan proses mengubah citra menjadi warna keabu-abuan. Setelah itu, lanjut ke tahap *otsu threshold* yang merupakan proses mengubah citra berderajat keabuan (*grayscale*) menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk objek dan *background* dari citra secara jelas. Kemudian, hasil dari kedua tahap tersebut akan dilakukan pengenalan citra oleh sistem. Hasil dari pengenalan tersebut berupa angka dan huruf dari plat nomor kendaraan.



Gambar 4 Flowchart Kerja easyOCR.

2.3.3. **Firestore**

Firestore menerima dan menyimpan data hasil kerja dari OCR dalam bentuk *string*. Kemudian data yang tersimpan akan dikirim ke aplikasi untuk menampilkan berapa persen tempat parkir yang masih tersedia. Setelah itu, aplikasi juga akan mengirimkan data *booking* yang sudah di-input oleh user ke firestore. Firestore akan mengecek data *booking* dan data yang dibaca oleh sistem OCR. Jika kedua data sudah sesuai, maka akan dikirim perintah ke ESP32. Jika kedua data tidak sesuai, maka sistem membaca user belum mem-*booking* tempat parkir. Setelah perintah dikirim ke ESP32, ESP32 meneruskan perintah ke servo untuk membuka palang secara otomatis. Sensor *proximity* akan mendeteksi kendaraan, jika sensor mendeteksi tidak terdapat kendaraan maka sensor akan mengirimkan perintah ke ESP32, kemudian ESP32 meneruskan perintah ke servo untuk menutup palang secara otomatis. Begitu pula, jika sensor mendeteksi terdapat kendaraan maka sensor tetap mengirimkan perintah ke ESP32, kemudian ESP32 meneruskan perintah ke servo yaitu untuk membuka palang terus - menerus.



Gambar 5. Flowchart Kerja Firestore.

3. Results and Analysis

Sistem pintu pengaman parkir otomatis ini bekerja saat pengendara kendaraan bermotor memasuki area parkir. Pada saat memasuki pintu parkir, terdapat kamera CCTV (dalam prototype digantikan oleh webcam) yang berfungsi untuk membaca plat nomor yang ada di kendaraan dan menyimpannya di dalam PC. Pada saat kendaraan di depan pintu portal sebelum melewati palang parkir, kamera mengambil gambar mobil yang selanjutnya diproses oleh algoritma YOLOv4 dengan custom model untuk mendeteksi lokasi nomor plat kendaraan pada gambar. Untuk pengujian dilakukan dengan 10 buah data gambar mobil pada lokasi parkir di siang hari dengan resolusi 960x744 pixel yang tertera pada Tabel 1.

Table 1. Hasil Deteksi YOLOv4.

No	Gambar Mobil	Scan Plat	Hasil Scan
1			

2			
3			
4			
5			
6			
7			
8			
9			
10			

Sistem YOLO berhasil mendeteksi posisi plat secara akurat dengan tingkat *confidence* rata-rata 98,75%. Dapat disimpulkan bahwa untuk pendeteksian plat sudah berjalan secara konsisten bahkan disaat *confidence threshold* dipasang di atas angka 80%.

Proses *preprocessing* tahap pertama dilakukan dengan memotong gambar plat kendaraan hingga $\frac{1}{3}$ dari total tinggi gambar plat dari arah bawah. Hal ini bertujuan untuk menghilangkan tanggal masa berlaku plat yang berada di bagian bawah plat sehingga sistem OCR dapat berjalan konsisten untuk mendeteksi nomor plat saja. Pada Gambar 6 dapat dilihat contoh pemotongan plat.



Gambar 6. Pemotongan plat untuk menghilangkan masa berlaku

Selanjutnya gambar plat di konversi menjadi grayscale dan dimanipulasi dengan *Otsu Threshold* untuk memisahkan *foreground* (tulisan nomor plat) dan *background* yang kemudian diubah kedalam bentuk *string*, seperti yang terlihat pada Tabel 2. *Preprocessing & Deteksi OCR*.

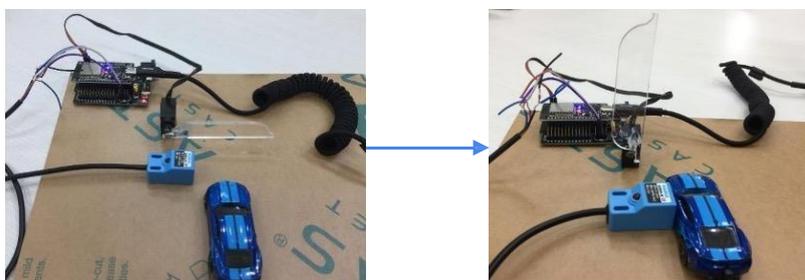
Table 2. *Preprocessing & Deteksi OCR*

Plat Awal	Proses Grayscale	Proses OtsuThreshold	Hasil	Akurasi
			DD1622M	100%
			DD1945KIY	100%
			B2384BKS	100%
			IDD1951R K	87,5%
			DD1367KV	100%
			DD8YT	100%
			DD1143MA	100%
			DD1510VS	100%
			DD1530SG	100%
			DD1548JE	100%

Akhirnya gambar yang telah dimanipulasi dilakukan proses pembacaan menggunakan OCR sebelum hasilnya dikirim ke *firebase* dan diteruskan ke ESP32 untuk melakukan perintah buka atau tutup palang.

Pada pengujian terdapat beberapa kegagalan deteksi yang diakibatkan oleh terdeteksinya garis pinggir plat yang berwarna putih sebagai karakter. Ditambah lagi pada kondisi di lapangan yang di mana jenis tulisan pada plat mobil terkadang tidak mengikuti standar sehingga menghasilkan hasil pembacaan yang kurang konsisten.

Palang parkir akan terbuka secara otomatis jika plat kendaraan yang dideteksi YOLOv4 dan proses OCR memiliki data yang sama dengan data booking yang di-input oleh user. Tingkat respon palang otomatis ditentukan dari kecepatan internet yang diterima. Simulasi parkir yang kami lakukan dapat dilihat pada Gambar 4.2.



Gambar 7. Simulasi Palang Parkir

4. Conclusion

Dari hasil pengujian sistem yang telah kami teliti didapatkan hasil yang memuaskan dari penggunaan YOLOv4 dan OCR untuk mendeteksi plat kendaraan dengan tingkat akurasi yang tinggi yaitu 98,75%. Proses pengiriman data yang dikirim ke dalam *firebase* menuju server sendiri tidak memerlukan waktu yang lama hanya saja kecepatan ini mungkin dapat berubah-ubah tergantung kecepatan jaringan internet yang tersedia di area parkir yang digunakan. Rata-rata waktu yang dibutuhkan untuk melakukan pengiriman setiap data ke dalam *firebase* adalah 3.0 detik.

Daftar Pustaka

- [1] I. Ghosh, "AIoT: When Artificial Intelligence Meets the Internet of Things," *Visual Capitalist*, Aug. 12, 2020. <https://www.visualcapitalist.com/aiot-when-ai-meets-iot-technology/> (accessed Oct. 02, 2022).
- [2] Y.-J. Lin *et al.*, "Artificial Intelligence of Things Wearable System for Cardiac Disease Detection," in *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, Mar. 2019. Accessed: May 11, 2023. [Online]. Available: <http://dx.doi.org/10.1109/aicas.2019.8771630>
- [3] W. C.-C. Chu, C. Shih, W.-Y. Chou, S. I. Ahamed, and P.-A. Hsiung, "Artificial Intelligence of Things in Sports Science: Weight Training as an Example," *Computer*, vol. 52, no. 11, pp. 52–61, Nov. 2019, doi: 10.1109/mc.2019.2933772.
- [4] C. Suardi, A. N. Handayani, R. A. Asmara, A. P. Wibawa, L. N. Hayati, and H. Azis, "Design of Sign Language Recognition Using E-CNN," in *2021 3rd East Indonesia Conference*

-
- on *Computer and Information Technology (EIconCIT)*, Apr. 2021. Accessed: May 11, 2023. [Online]. Available: <http://dx.doi.org/10.1109/eiconcit50028.2021.9431877>
- [5]]Republika, "Potensi Tinggi Implementasi IoT," *Republika.id*, 2022. <https://www.republika.id/posts/24451/potensi-tinggi-implementasi-iot%C2%A0> (accessed May 11, 2023).
- [6] K. Guo, Y. Lu, H. Gao, and R. Cao, "Artificial Intelligence-Based Semantic Internet of Things in a User-Centric Smart City," *Sensors*, vol. 18, no. 5, p. 1341, Apr. 2018, doi: 10.3390/s18051341.
- [7] Y. Yin, W. Zhang, S. Hong, J. Yang, J. Xiong, and G. Gui, "Deep Learning-Aided OCR Techniques for Chinese Uppercase Characters in the Application of Internet of Things," *IEEE Access*, vol. 7, pp. 47043–47049, 2019, doi: 10.1109/access.2019.2909401.
- [8] L. A. Glasenapp, A. F. Hoppe, M. A. Wisintainer, A. Sartori, and S. F. Stefenon, "OCR Applied for Identification of Vehicles with Irregular Documentation Using IoT," *Electronics*, vol. 12, no. 5, p. 1083, Feb. 2023, doi: 10.3390/electronics12051083.
- [9] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, 2020, doi: 10.1109/access.2020.3012542.
- [10] S. F. Stefenon, K.-C. Yow, A. Nied, and L. H. Meyer, "Classification of distribution power grid structures using inception v3 deep neural network," *Electrical Engineering*, vol. 104, no. 6, pp. 4557–4569, Sep. 2022, doi: 10.1007/s00202-022-01641-1.